REMARKS/ARGUMENTS

<u>Obviousness rejection of Claim 1</u>

Claim 1 stands rejected under 35 U.S.C. §103 as being obvious over the teachings of US Patent 7,080,081 hereinafter Agarwal, in view of US Patent 5,517,631 hereinafter Machado. This rejection is respectfully requested to be withdrawn for several reasons as follows.

In explaining the rejection of Claim 1, the Examiner stated as follows in the middle of page 5 of the above-identified Office Action (original emphasis):

> As per **claim 1,** Agarwal et al. discloses **a method implemented in a computer, the method comprising:**
> **generating an array update operation based on a query to update a relational database; wherein said array update operation specifies a plurality of row-identifier and value pairs to update multiple rows in a table of said relational database;**
> as (see e.g., col. 2 lines 39-48).

Note that in the above-quoted explanation, the Examiner has only provided a single citation, without providing any indication as to what item in Agarwal is being analogized to a corresponding item in Applicants' Claim 1. Accordingly, the Examiner is respectfully requested to identify a specific step in the method disclosed by Agarwal that is analogized to Claim 1's "generating." Moreover, the Examiner is respectfully requested to identify a specific operation disclosed by Agarwal that is analogized to Claim 1's "array update operation."

For convenience, the Examiner-cited text from Agarwal is reproduced below:

> According to yet another aspect of the invention, the method further includes the step of processing a query for information stored in the table. According to yet another aspect of the invention, processing a query further includes using information from either the individual block indexes or the composite index to obtain a list of block identifiers, and

scanning blocks of the table for records. According to another aspect of the invention, processing a query includes the steps of scanning the entire table for records and using a record-based index to find records.

The above-quoted text from Agarwal discloses processing a query in two alternative ways as follows: (1) to obtain a list of block identifiers and (2) to find records by scanning the entire table. There appears to be no further disclosure, in the above-quoted text, to generate an operation as expressly recited in Claim 1.

Specifically, as per the first alternative way (1) noted above, Agarwal discloses obtaining a list of block identifiers, but Agarwal does not disclose generating an array update operation that specifies a plurality of row-identifier and value pairs.

In this context, Applicants respectfully traverse the Examiner's statement (later in page 5 of the Office Action) "the records interpreted as a group of row-identifier and value pairs (col. 2, lines 46-48)." In response, Applicants note that Claim 1 does not recite "records" generically. Instead Applicants' Claim 1 recites more specifically "row-identifier and value pairs." The level of detail recited in Claim 1 cannot be ignored by the Examiner even under the broadest reasonable interpretation.

The explicit language in Claim 1, namely "row-identifier and value pairs" distinguishes over Agarwal's records for the following reasons. Agarwal's blocks are organized by value. For example, all records in Agarwal's block 1 shown in Agarwal's FIG. 5 have the value "9901" for YearAndMonth and the value "AB" for Province. Hence, Agarwal discloses values at block level, and not values paired with records. As multiple records in Agarwal's block are disclosed to have the same value, there appears to be no need in Agarwal's method to form pairs at the record level.

In this context, note that the Examiner cited to Agarwal's column 6, lines 43-50 at the bottom of page 5 of the Office Action. For convenience, the Examiner-cited text from Agarwal is reproduced below:

In the exemplary multidimensional table depicted in FIG. 4, to find the slice containing all records with `ON` for the Province

> dimension, we would look up this key value in the Province
> dimension block index, and find a key such as the following:
> <ON: 9, 16, 18, 19, 22, 24, 25, 30, 36, 39, 41, 42>
>
> where the key is in the form of a <key value: BID(s)>pair.

Thus Agarwal discloses a single "key value" paired with multiple block identifiers. Note that each block in Agarwal can hold multiple records. Hence, Agarwal discloses a single value for multiple records. In contrast, Applicants' value is paired with each row-identifier, as recited in Claim 1 "wherein each value comprises data to be updated in said row identified by said row-identifier."

Moreover, Agarwal expressly discloses processing a single block at a time from a list, as stated at column 8, lines 40-41. Agarwal states "This would involve just one I/O as a block is stored as an extent on disk and can be read into the bufferpool as a unit." So, Agarwal appears to be teaching away from generating an array update operation that specifies multiple row-identifier and value pairs, as recited in Claim 1.

Also, as per the second alternative way (2) noted above, Agarwal discloses finding records by scanning the entire table, but Agarwal does not appear to indicate that the multiple records found in this manner are specified to be updated to multiple values paired thereto. Instead, it appears that in scanning the entire table, Agarwal finds multiple records that all have the same value (on which the scanning is done). Therefore, even in the second alternative way (2), there appears to be no rationale to change Agarwal's method to form pairs at the record level, and additionally generate an array update operation for the pairs as recited in Claim 1.

In summary, Applicants respectfully submit that there appears to be no disclosure in Agarwal to perform "generating" as recited in Claim 1.

In modifying the teachings of Agarwal with Machado, at most a skilled person may use Machado's "single sequence" to retrieve information from a single block of Agarwal, because as noted above Agarwal appears to teach retrieving a single block at a time.

Furthermore, note that Agarwal's value is already pre-existing, i.e. Agarwal's value cited against Claim 1 is an old value. In contrast, Claim 1 expressly states that each value comprises data to be updated, i.e. new data. Hence, Agarwal fails to disclose row-identifier and value pairs in which each value comprises data to be updated as recited in Claim 1. Therefore, this is an additional distinction of Claim 1 over Agarwal.

Moreover, even if Machado were used to retrieve multiple blocks identified by Agarwal as proposed by the Examiner, only those multiple blocks are retrieved that have the same value in a single slice of Agarwal's multidimensional table. See Agarwal's column 6, lines 43-50 cited by the Examiner. Hence, generating an array update operation that specifies multiple values as recited in Claim 1 (in the group of row-identifier and value pairs) appears to not be disclosed in the combined teachings of Agarwal and Machado.

Additionally, in explaining the rejection of Claim 1, the Examiner stated in the middle of page 6 of the Office Action, that Claim '1s limitation on "repeatedly updating" is disclosed as Insert, Delete, Purge and Update functions in Agarwal's FIG. 8, and col. 11, line 4 to col. 12 line 55. This explanation is respectfully traversed because the Examiner has not fully explained their position. Specifically, the Examiner cited Agarwal's Update function can be analogized to Claim 1's "updating," and if this interpretation is correct, the record is not clear as to why is the Examiner additionally citing Insert, Delete and Purge functions of Agarwal? Moreover, Agarwal's Update function is of two kinds, namely "Simple update" and "Update of the dimension columns" as stated by Agarwal in column 12, lines 37 and 48. It is unclear from the Examiner's remarks as to which kind of update function of Agarwal is being cited against Claim 1's updating.

Assuming the Examiner is citing "Update of the dimension columns" of Agarwal, note that Agarwal's update is not of the same blocks. Specifically, when the value of multiple records is to be updated, those multiple record(s) are deleted from an old cell and inserted in a new cell of Agarwal's multidimensional table. See Agarwal's column 12, lines 50-53. Agarwal's disclosure of "old" and "new" cells teaches away from

Applicants' repeatedly updating, in blocks which are in the cache, each row identified in the group of row-identifier and value pairs, using a corresponding value in the row-identifier and value pairs as recited in Claim 1.

For the above-discussed reasons, reconsideration and withdrawal of the obviousness rejection of Claim 1 is respectfully requested. Claims 2-11, 19 and 21 depend from Claim 1 and are, therefore, likewise patentable for at least the above-discussed reasons in reference to Claim 1.

Claims 2, 3 and 19

The Examiner's explanation of the rejection of Claim 2 is respectfully traversed because the Examiner's citations do not disclose any "sorting" as recited in Claim 2. Specifically, the Examiner cited to col. 5, lines 60-62 and col. 6, lines 10-13 of Agarwal in the middle of page 7 of the Office Action. As seen from the following quotation from Agarwal, there is no "sorting" in this Examiner-cited text:

> In the diagram, a block is represented by an oval, and is numbered according to the logical order of allocated extents in the table.
>
> ...
>
> While blocks are numbered sequentially starting from block 1 in the exemplary table shown herein, it should be appreciated that the blocks could be identified in numerous other ways.

The Examiner additionally cited to Machado's column 5, lines 33-40, but this citation also fails to disclose "sorting" as recited in Claim 2, as seen from the following quotation of the Examiner-cited text from Machado:

> The programmable data sequencer comprises a writeable control store including a random access memory area directly addressable by the programmed digital microcontroller for

writing sequences of control patterns, there being most preferably a single sequence written for controlling states of the programmable data sequencer during both data read and data write operations to and from the disk surface and a buffer memory.

Finally, at the bottom of page 7 of the Office Action, the Examiner cited to Machado's column 3, line 63 to column 4, line 1, which is reproduced below:

A more specific object of the present invention is to provide a data sequencer for a disk drive employing zoned data recording having data fields split into segments by intervening embedded servo sectors and wherein the data sequencer provides for automatic sequencing of data blocks during writing data to, and reading data from, the split data fields, in a manner which overcomes limitations and drawbacks of the prior art approaches.

As seen from the above-quoted text, there is no "sorting" whatsoever. Accordingly, Applicants respectfully submit that the Examiner's explanation fails to show that the combination Agarwal and Machado disclose sorting block-identifiers, prior to retrieval of the blocks as recited in Claim 2.

Claim 3 provides a further distinction over Agarwal and Machado by reciting that the sorting is performed subsequent to storage of the block-identifiers in the structure. In explaining the rejection of Claim 3, the Examiner cited to Agarwal's column 2 lines 42-48, column 5 lines 60-62 and column 6 lines 10-13. However, nothing in this Examiner-cited text by Agarwal appears to disclose sorting "subsequent to" storage of block identifiers in the structure. Claim 3 is therefore patentable for this additional reason.

Claim 19 further distinguishes over the combination Agarwal, Machado and Gold by requiring Claim 2's sorting to be based on adjacency of blocks relative to one

another. Claim 19's "adjacency" appears to be not disclosed Gold's paragraphs 0028 and 0031, which are reproduced below:

[0028] The reordered file list provided by sorter 220 gives the files in the order in which they are to be found on the disk 100. The sequence of the sectors on disk 100 begins with a nominated sector on the outermost track and proceeds anti-clockwise around the entire track. The sequence continues with the next track in and proceeds around that track in an anti-clockwise manner similar to that for the first track. The sequence continues from track to track in this way until the final sector is reached on the innermost track.

[0031] It will be appreciated that, due to the action of sorter 210, the files are not read from the disk in alphabetical order but in order of their physical locations on the disk, i.e. Y, X . . . , Z, D, A, G, H . . . , E, F, B, N, O . . . , L, M, C . . . The order in which the files have been retrieved and written to the back-up storage is logged in order to facilitate swift recovery of files from the tape unit 216. The reduction of the delays in the reading of data from disk 100 means that the tape drive 216 receives the back-up data at a rate high enough to prevent the tape drive 216 from pausing during writing of data to the tape. If tape drive 216 pauses, then it will switch off until the data steam resumes. By reducing the delays in reading data from the disk, the likelihood of the tape drive 216 switching off during back-up is reduced. This helps to shorten the back-up procedure since, if the tape drive 216 turns off then there is a significant delay whilst it starts up following a pause in the datastream from writer 214.

Claims 4, 5, 7, and 8

Claims 4, 5, 7 and 8 were rejected in the current office action based on the same citations as those provided by the Examiner in the prior Office Action dated March 12, 2008. Hence, Applicants respectfully draw the Examiner's attention to Applicants' arguments in support of Claims 4, 5, 7 and 8 as submitted in the Amendment dated July 14, 2008.

Claims 13-18 and 20

Claims 13-18 and 20 recite one or more limitations that are supported by arguments for patentability that are similar to one or more of the arguments presented above in reference to Claim 1. Accordingly, these claims are also similarly patentable.

Conclusion

Hence, Applicants respectfully request allowance of all pending claims. Please call the undersigned at (408) 378-7777 ext 113 in case of questions.

CERTIFICATE OF EFS-WEB TRANSMISSION
I hereby certify that this correspondence is being electronically transmitted to the U.S. Patent and Trademark Office to via the USPTO Electronic Filing System on the below date.

/omkar suryadevara/	October 22, 2009
Attorney for Applicant(s)	Date of Signature

Respectfully submitted,

/omkar suryadevara/

Omkar K. Suryadevara
Attorney for Applicant(s)
Reg. No. 36,320